

# Ein autonomes, mobiles Multi-Robotersystem für Intralogistik in der Produktion

**Lars Kistner**, Axel Dürrbaum, Andreas Kroll  
Fachgebiet Mess- und Regelungstechnik, Universität Kassel

## **Kurzfassung**

Der Artikel adressiert den Einsatz kooperierender mobiler Multi-Robotersysteme in der Intralogistik. Die Umsetzung erfolgt mittels Erweiterung des in der Wissenschaft verbreiteten Software-Frameworks ROS (Robot Operating System). Der Standardfunktionsumfang von ROS wurde zur Unterstützung des Multi-Roboterbetriebs erweitert, um Navigationsaufgaben wie das Passieren von Engstellen im robotischen Frontalverkehr meistern zu können. Das Multi-Robotersystem wurde in der Modellfabrik „µPlant“ des Fachgebiets Mess- und Regelungstechnik der Universität Kassel implementiert, um alle intralogistischen Aufgaben zu lösen. Im Beitrag werden Aufbau und Funktionen des mobilen Multi-Robotersystems und Erfahrungen aus dem bisherigen Testbetrieb vorgestellt.

## **1. Motivation**

Für den flexiblen Warentransport in der Produktion werden bislang meist fahrerlose Transportsysteme (FTS) eingesetzt [11]. Mit der zunehmenden Flexibilisierung der Produktion bis zur „Losgröße 1“ ergeben sich Anforderungen an die Intralogistik, die mit herkömmlichen Materialflusslösungen nur schwer zu erfüllen sind. Deshalb wächst das Interesse, autonome mobile Roboter (AMR) für diese Aufgaben einzusetzen, die leistungsfähigere Autonomie- und Navigationsfunktionen aufweisen.

### **1.1 FTS in der Intralogistik**

Bei fahrerlosen Transportsystemen handelt es sich definitionsgemäß [13] um flurgebundene Fördermittel mit eigenem Fahrantrieb, welche automatisch gesteuert und berührungslos geführt werden. Ziel ist dabei ein flexibler Warentransport ohne größere Eingriffe in die Umgebung, wie sie für Förderbänder oder ähnliches notwendig sind. Ein FTS agiert zumindest teilweise im selben Bereich wie Menschen; deshalb ist ein Personenschutz notwendig. FTS orientieren sich an Spuren, meist Farbstreifen auf oder Induktionsschleifen im Boden, und an Haltesignalen. Durch das Spursystem kann auf eine aufwendige Wegplanung und Selbstlokalisierung verzichtet werden. Dies nimmt dem System jedoch auch die Möglichkeit, bei versperrtem Weg alternative Routen zu wählen, um seinen Betrieb selbständig fortzusetzen.

Die Entwicklung von FTS verlief in vier Epochen [15]: Der Beginn datiert auf 1954 mit induktiver Spurführung und Start/Stopp-Markierungen. 1970-1990 gab es Bordrechner in Schaltschränken mit Datenübertragung via Infrarot/Funk. 1990-2010 erfolgte der Einsatz von speicherprogrammierbaren Steuerungen und Mikro-Computern und es gab eine „freie“ Navigation mit Magnetfeldern oder Laser-Triangulation. Seit 2010 erfolgte eine Annäherung an autonome mobile Roboter und Außeneinsätze wurden möglich. Üblicherweise setzen FTS für die Kollisionsvermeidung auf taktile Sensoren und Laserscanner mit unterschiedlichen Alarmzonen. Durch den zunehmenden Einsatz von Digitaltechnik nahmen die Fähigkeiten bezüglich Navigation und Autonomie zu.

## **1.2 Autonome Mobile Roboter**

Ein klassischer Roboter in der Produktion ist ein frei und wieder programmierbarer, multifunktionaler Manipulator mit mindestens drei unabhängigen Achsen, um Materialien, Teile, Werkzeuge oder spezielle Geräte auf programmierten, variablen Bahnen zu bewegen, um so verschiedenste Aufgaben zu erfüllen [14]. Ein mobiler Roboter kann sich (im Gegensatz zum FTS) frei in seiner Umgebung bewegen. Autonome mobile Roboter (AMR) zeichnen sich dadurch aus, dass sie ohne physische Daten- und Energieübertragung (z. B. durch Kabel) agieren können. Dazu muss ein AMR über Möglichkeiten zur Navigation und der dazu notwendigen Selbstlokalisierung verfügen [10]. AMR benötigen keine Spuren; sie navigieren auf der gesamten freien Fläche und reagieren im Rahmen ihrer Möglichkeiten auf Probleme wie z.B. ungeplante Hindernisse. So gibt es Parallelen zu autonomen Fahrzeugen im öffentlichen Straßenverkehr.

## **1.3 Besondere Anforderungen in der Intralogistik**

Unter Intralogistik versteht man den Material- und Warentransport innerhalb der Produktion. Dabei gibt es einige Besonderheiten, die beim Einsatz mobiler Roboter zu beachten sind: Meist findet der Transport auf ebenen Flächen in einer bekannten, aber sehr engen Umgebung statt. Dies erschwert Aufgaben wie das sichere Passieren entgegenkommender Roboter an Engstellen. Oft treten unvorhergesehene Hindernisse wie z.B. verlorenes oder ungeplant abgestelltes Material auf. Innerhalb der Produktion bewegen sich zudem Menschen und von ihnen gefahrene Fahrzeuge, deren Bewegungen nur schwer vorherzusagen sind.

Intralogistische Aufgaben werden meist in geschlossenen Räumen ausgeführt, was zwar den Einfluss von Umweltfaktoren vermindert, aber auch die Selbstlokalisierung mit GPS, die Standard im Außenbereich ist, fast immer unmöglich macht. Zudem gibt es zahlreiche Störeinflüsse auf die drahtlose Kommunikation, wie Verschattungen, Störstrahlung durch Maschinen und Koexistenzprobleme mit anderen Funksystemen in der Produktion. Große Schwierigkeiten für die Lokalisierung bereiten merkmalsarme Umgebungen wie Flure. „Merkmalsarm“ bedeutet in

diesem Zusammenhang, dass unterschiedliche Positionen ein gleiches oder zumindest sehr ähnliches Sensorbild liefern.

In diesem Beitrag wird ein autonomes, mobiles Multi-Robotersystem vorgestellt, das in der Modellfabrik  $\mu$ Plant [6,7] umgesetzt wurde. Im Gegensatz zu reinen Lageranwendungen wie bei Amazon [2] oder Alibaba [8] erfolgt der Einsatz der Roboter in einer Produktionsanlage. Das System sorgt für den automatischen Produkttransport nicht nur zwischen Lagerorten, sondern auch zwischen den einzelnen Produktionseinrichtungen.

## 2. Roboterplattform & Infrastruktur

Das für die Intralogistik in der Modellfabrik  $\mu$ Plant eingesetzte Multi-Robotersystem besteht aus vier baugleichen kostengünstigen mobilen Robotern (Abb.1) und einem Server (Desktop-PC) zur Aufgabenplanung und -verteilung.

### 2.1 Basisplattform TurtleBot2

Die Basis für die mobilen Roboter ist die Roboter-Plattform „Kobuki“ von Yujinrobot [5]. Sie hat einen Durchmesser von 0,35 m und erreicht bei Geradeausfahrt maximal 0,65 m/s. Die Positionsbestimmung erfolgt maßgeblich über Umdrehungszähler in den beiden angetriebenen Rädern des Differentialantriebs und einen Drehratensensor ( $\dot{\varphi}_{\max} = 110^\circ/\text{s}$ ) zur Ermittlung der Ausrichtung. Um Kippstabilität zu gewährleisten, verfügt jeder Roboter zusätzlich über zwei Castorräder. Eine Microsoft Kinect Tiefenkamera unterstützt die Lokalisierung und ermöglicht das Erkennen von Hindernissen. Zusätzlich stellt sie ein Live-Videobild zur Verfügung. Ein auf dem Roboter montierter Mini-PC dient zur Steuerung und Kommunikation. Für den Materialtransport verfügt jeder Roboter über eine Aufnahme für zwei verschieden große Transportbecher (Abb. 2). Die Kombination dieser Roboter-Plattform mit Tiefenkamera und einem Onboard-PC wird als „TurtleBot2“ [12] bezeichnet. Er wiegt insgesamt (Roboter, Aufbauten, Sensoren und PC) 5,8 kg und kann eine Nutzlast von maximal 3,5 kg transportieren. Die Positioniergenauigkeit ist üblicherweise besser als 0,15 m. Außerdem verfügt die Roboter-Plattform über drei Infrarot-Sensoren, um über Leitstrahlen ein Roboter-Dock exakt anfahren zu können, da die verwendete Tiefenkamera dort nicht ausreichend genau funktioniert: Die Kinect erfordert einen Mindestabstand von etwa 0,5 m und arbeitet bis zu einem Maximalabstand von

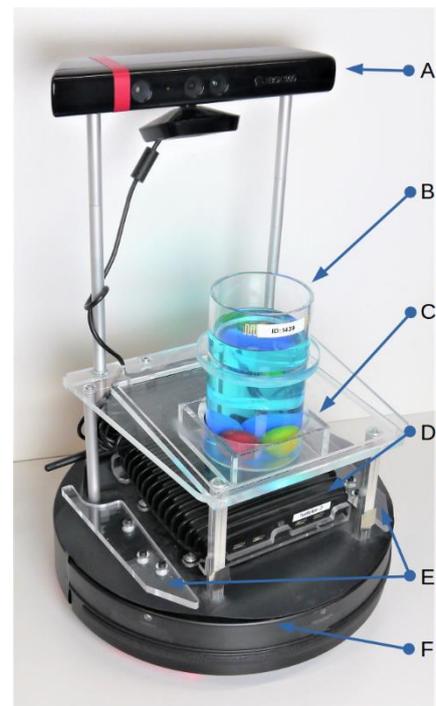


Abbildung 1: TurtleBot2  
(A) Tiefenkamera; (B) kleiner Becher mit Produkt; (C) Transporthalterung; (D) Mini-PC; (E) mechanische Führungsschienen; (F) Roboterplattform

ca. 8,0 m vom Zielobjekt. Im Gegensatz zu häufig eingesetzten 2D-Laserscannern liefert die Kinect Tiefenbilder. Sie kann nur im Innenbereich eingesetzt werden, da die verwendete IR-Projektion bei größerer Helligkeit überstrahlt wird.

## 2.2 Rechner und Kommunikation

Die Datenverarbeitung und Wegplanung erfolgt bei jedem Roboter durch einen passiv gekühlten Mintbox2 Mini-PC mit Intel i5 Prozessor. Diese Mini-PCs haben sich bereits in den Inspektionsrobotikprojekten Robot}AIR{ [4] und RoboGasInspector [3] bewährt. Der Datenaustausch der Roboter untereinander und mit dem Server-PC erfolgt über WLAN. Durch den 2013 veröffentlichten WLAN Standard IEEE 802.11ac kann die Kommunikation mit preiswerten Standardkomponenten erfolgen. Das weniger genutzte 5 GHz-Band und die Mehrkanal-Technik (MU-MIMO) ermöglichen eine höhere Bandbreite bei geringen Datenpaketlaufzeiten auch bei hoher Kanalbelastung. Dies ist für die Koordination der Roboter untereinander und die Überwachung wichtig. In der Modellfabrik ist eine WLAN-Datenrate von bis zu 867 Mbit/s möglich.

## 2.3 Nutzlastkonzept

Die Aufgabe der mobilen Roboter besteht im Transport von Flüssigkeiten und der Bereitstellung von Produktgebinden. Dazu werden kleine Acryl-Becher für bis zu 500 ml und große für bis zu 2500 ml verwendet (Abb. 2). Die großen Becher dienen dem Vorprodukttransport von einer Prozessinsel zur nächsten. Die kleinen Becher dienen dem Produkttransport und sind zudem auch das Gebinde des Endprodukts bestehend aus Flüssigkeit und Aromakugeln. Sie können im Hochregallager eingelagert und in einer Re-



Abbildung 2: großer Becher (l.); kleiner Becher (r.)

ycling-Station geleert werden. Die Roboter sind gegen in Folge von Leckagen, Überschwappen o.ä. auf sie gelangende Flüssigkeit geschützt: Diese kann ablaufen, ohne an kritische Punkte zu gelangen. Jeder Becher ist mit einem RFID-Tag bestückt. Das ermöglicht in den Roboter-Docks eine Überprüfung, ob ein Becher vorhanden ist und falls dies der Fall ist, welcher Bechertyp und welches Produkt vorliegen.

## **2.4 Energieversorgung**

Der Roboter und der Onboard-Mini-PC werden über einen Lithium-Ionen-Akku mit 4,4 Ah versorgt, der je nach Belastung einen autonomen Betrieb von ca. 120 bis 165 Minuten ermöglicht. In jedem Roboter-Dock einer Produktionseinrichtung oder des Hochregallagers ist eine elektrische Ladestation integriert, die den Roboter-Akku während des Aufenthaltes dort etwas aufladet. Liegen einem Roboter keine Transportaufträge vor, so fährt er zu einer der drei separaten elektrischen Ladestationen, um keine Produktionseinrichtung zu blockieren. Die Zeit für eine vollständige Ladung hängt stark vom aktuellen Stromverbrauch des Mini-PCs ab und schwankt zwischen 2 (Onboard-PC im Leerlauf) und 5 Stunden (Onboard-PC unter Volllast). Im Testbetrieb waren keine Extrapausen für das Aufladen notwendig.

## **2.5 Sonstige Merkmale**

Die vier Roboter sind in Soft- und Hardware identisch aufgebaut. Zwecks Unterscheidbarkeit ist jeder Roboter mit RGB-LEDs ausgestattet, welche je nach Softwarekonfiguration eine andere Farbe annehmen. Das erleichtert die Zuordnung der Roboter zu ihren digitalen Abbildern in der grafischen Nutzeroberfläche und die manuelle Fernsteuerung im Fall von Störungen oder Tests. Zudem können die LEDs als Zustandsanzeige genutzt werden: Da die mobilen Roboter keine Displays haben ist eine Ermittlung ihres Status bis zum Hochfahren des Betriebssystems und automatischer Anmeldung im Netzwerk schwierig.

## **3 Robot Operating System**

Die Software der Transportroboter baut auf das Robot Operating System (ROS) auf. ROS ist die bedeutendste Open-Source Software-Sammlung für die autonome mobile Robotik.

### **3.1 Historische Entwicklung**

Die Entwicklung des Robot Operating Systems (ROS) wurde 2007 am Stanford Artificial Intelligence Laboratory begonnen. ROS hat sich bis heute zum bedeutendstem Rahmenwerk für die mobile Robotik im akademischen Bereich entwickelt. Die Entwicklung unter Open-Source-Lizenz, die Pflege von Treibern für viele Sensoren und Roboterplattformen, die Bereitstellung von Softwarepaketen für Standardaufgaben der Robotik und ein einheitliches Nachrichtensystem sorgen für eine ausgeprägte Modularität. Diese verhindert, dass Aufgaben immer wieder von neuem gelöst werden müssen. ROS erfährt eine sehr aktive Entwicklung: In den letzten Jahren gab es jährlich eine neue Hauptversion, von denen einige größere Änderungen mit sich brachten. Mit der Veröffentlichung der ersten Version des „ROS 2.0“-Zweigs im Dezember 2017 wurde der Fokus verstärkt auf neue Ziele gelenkt. Dazu gehören u. a. Multi-Robotersysteme,



## 4 Einzelrobotersystem

Dieser Abschnitt behandelt die grundlegenden Aufgaben, die ein AMR-System lösen muss.

### 4.1 Selbstlokalisierung

Für die Selbstlokalisierung sollte die initiale Pose (Position und Ausrichtung) eines Roboters bekannt sein. Im Freien kann dies durch GPS erfolgen, im Innenbereich bieten sich Marker (bspw. QR-Codes in den Ladestationen der Roboter) an. Die weitere Abschätzung der Position erfolgt über die Fusion der Odometrie-Daten mit denen anderer Sensoren. Häufig werden Laserscanner oder Tiefenkameras eingesetzt. Um aus einem Laserscan eine Positionsschätzung zu ermitteln, ist eine digitale Karte erforderlich. Man sucht dann die Pose in der Karte, die die Tiefenmessung des Sensors bestmöglich erklärt. Um den Aufwand zu reduzieren, erfolgt die Suche nur in einer lokalen Umgebung. Deshalb ist eine hinreichend genaue Kenntnis der initialen Position und Ausrichtung wichtig. Theoretisch kann mit dem SLAM-Verfahren (Simultaneous Localization and Mapping) eine Karte während des laufenden Betriebs erstellt werden. Das ist zum Beispiel bei Such- und Rettungsmissionen im unbekanntem Terrain sinnvoll. In einer definierten, frequentiert genutzten Umgebung wie im Fall der Intralogistik sollte eine Karte vor dem Produktionseinsatz der Roboter mit SLAM-Verfahren erzeugt werden. Die Genauigkeit der Karte hängt maßgeblich von den eingesetzten Sensoren und der Fahrgeschwindigkeit ab. Eine Nachbearbeitung der Karte ist meist sinnvoll. Abb. 8 zeigt eine unbearbeitete, per SLAM erzeugte Karte. Deutlich sind die verrauschten und nicht immer ganz geraden Wände zu erkennen.

### 4.2 Wegplanung

Die Wegplanung erfolgt auf zwei Ebenen. Die **strategische** Wegplanung erfolgt von der Ziel- zur Istposition anhand der Karte mit bspw. einem Raster von 15 cm x 15 cm. Dabei wird die Annahme getroffen, dass die Karte alle relevanten Hindernisse enthält. Die **taktische** Wegplanung sucht für die nächsten zwei Meter einen für den Roboter fahrbaren Weg, der sich an der strategischen Wegplanung orientiert. Dabei wird ein feineres Raster verwendet, bspw.

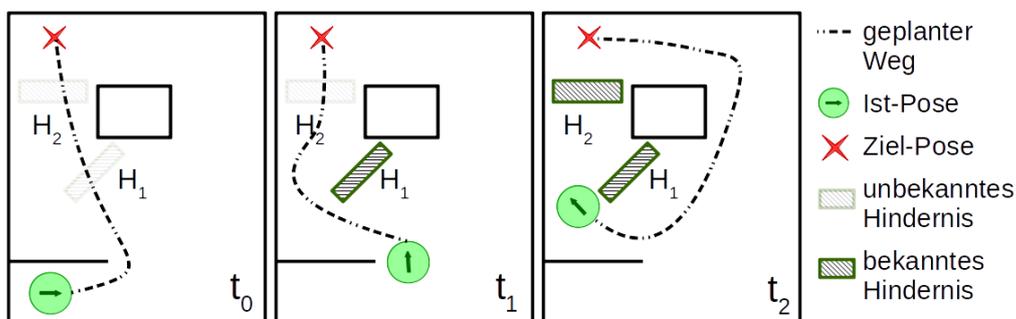


Abbildung 4: Typischer Ablauf einer Wegplanung: der strategische Plan zum Zeitpunkt  $t_0$  muss durch unvorhergesehene Hindernisse erst angepasst ( $t_1$ ) und dann komplett neugeplant ( $t_2$ ) werden

1 cm x 1 cm. Von der Sensorik erkannte, aber nicht kartierte Hindernisse werden in die Karte aufgenommen und bei erneuter strategischer und taktischer Wegplanung berücksichtigt. Abb. 4 zeigt exemplarisch den Ablauf einer Wegplanung. Zum Zeitpunkt  $t_0$  (Abb. 4 links) erfolgt die strategische Wegplanung anhand der vorliegenden Karte und die kürzeste Route wird gewählt. Im Einsatzes wird zum Zeitpunkt  $t_1$  (Abb. 4 Mitte) vom Roboter das Hindernis  $H_1$  erkannt. Es kann umfahren werden und die Route wird entsprechend neu geplant. Zum Zeitpunkt  $t_2$  (Abb. 4 rechts) wird das Hindernis  $H_2$  erkannt, welches den geplanten Weg vollständig versperrt. Die Route muss komplett geändert bzw. neu geplant werden.

Die Einführung von zwei Planungsebenen (taktisch & strategisch) mit unterschiedlichen Detaillierungsgraden bringt Vorteile. Eine grobe Auflösung bei der globalen Wegplanung hält den Rechenaufwand für die i.d.R. lange Route gering. Für die lokale Planung in der direkten Umgebung der Roboter wird mit einer hohen Auflösung gearbeitet, um bspw. Engstellen kollisionsfrei meistern zu können. Der ROS-Navigation-Stack verwendet standardmäßig den Dijkstra-Algorithmus für die Wegplanung.

### 4.3 Kollisionsvermeidung und Hindernisumfahrung

Zur Wegplanung und Kollisionsvermeidung wird ein kostenbasierter Ansatz verfolgt. Startpunkt ist die mittels SLAM erzeugte 2D-Karte (siehe Abschnitt 4.1). Ihr wird eine zweite Karte überlagert, die alle im Einsatz erfassten Hindernisse enthält. Die resultierende Gesamtkarte wird bspw. in ein Raster von 1 cm x 1 cm großen Zellen diskretisiert. Für jede Zelle wird der minimale euklidische Abstand zu den Hindernissen ermittelt. Bei der Wegplanung wird die harte Einhaltung eines Mindestabstandes gefordert. Für Abstände größer als der Mindestabstand wird einer Zelle ein exponentiell mit dem Abstand fallender Kostenterm zugewiesen. Dann wird derjenige Pfad zwischen Ziel- und Startpunkt gewählt, der zu minimalen Gesamtkosten führt. In Abb. 5 sind zwei Routen zu sehen. Je nach gewählter Kostenfunktion wird die kurze aber näher an Hindernissen gelegene Route 1 oder aber die längere Route 2 gewählt. Hindernisse werden stets als statisch angesehen. Menschen und Fahrzeuge werden daher, falls sie sensorisch erfasst werden, immer wieder an ihrer aktuellen Position temporär als statisches Hindernis in der Karte eingetragen. Eine Abschätzung ihrer Bewegungsrichtung und Geschwindigkeit erfolgt nicht. Diese Schwäche verhindert

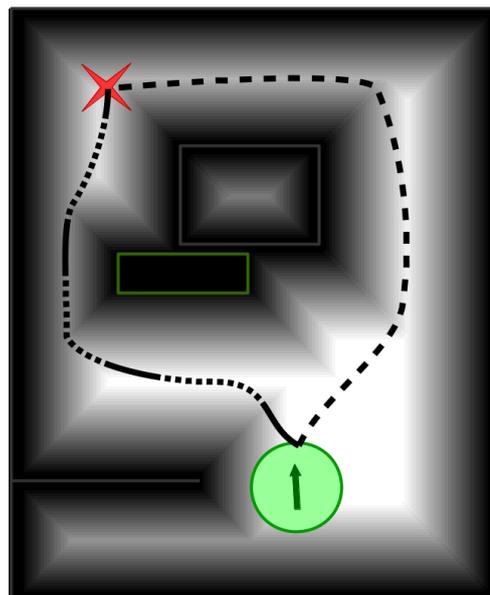


Abbildung 5: Kostenkarte mit Route 1 (---) und Route 2 (.....)

einen flüssigen Multi-Roboterbetrieb, weshalb in Abschnitt 5 Verbesserungen eingeführt werden.

#### 4.4 Systemüberwachung

Ein Roboter steuert und überwacht sich selbst. Das zentrale Planungssystem, welches die Roboter zu den Zielen schickt, muss also nicht die einzelnen Fahrten überwachen. Es muss jedoch gelegentlich prüfen, ob es Rückmeldungen zu den Fahraufträgen gibt, und bei Problemen darauf reagieren. Übliche Rückmeldungen sind „Ziel erreicht“, „Navigation abgebrochen“ oder „Fahre zu Ziel“. Bleibt eine Rückmeldung aus, so ist die Verbindung unterbrochen. Basierend auf den Rückmeldungen werden Maßnahmen ergriffen, dass der Roboter das Ziel erreicht oder, falls nicht möglich, ihn in einen definierten Zustand bringt (bspw. indem er in eine elektrische Ladestation fährt). Letzte Entscheidungsinstanz zum Beheben von Fehlern ist der Mensch. Zur Unterstützung bietet ROS die grafische Visualisierung RViz an (Abb. 6).

### 5 Erweiterungen zum Multi-Robotersystem

Bei der gleichzeitigen Verwendung mehrerer Roboter ergeben sich neue Anforderungen bzgl. Navigation und Aufgabenverteilung.

#### 5.1 ROS und Multi-Roboterunterstützung

ROS ist vom Konzept her Multi-Roboter fähig. Insbesondere die Gruppierung in Namensräume ermöglicht eine saubere Trennung gleichartiger Systeme. Ein Multi-Robotersystem war bisher

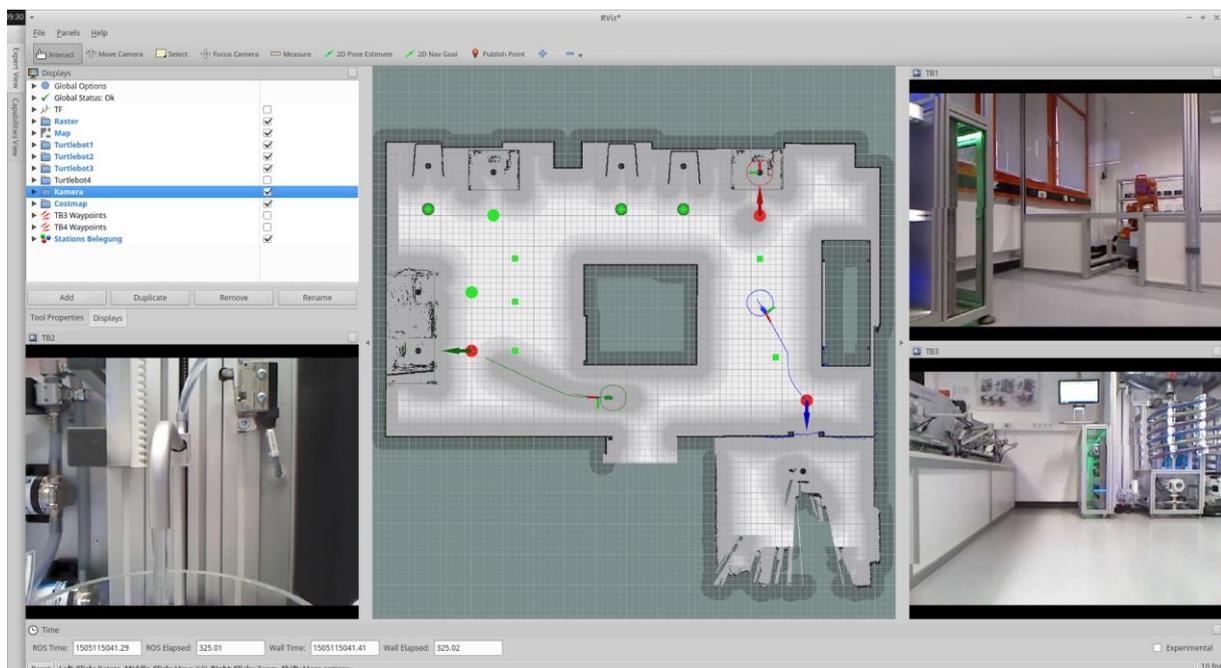


Abbildung 6: RViz: Grafische Visualisierung von ROS in der Modellfabrik: Digitale Karte (Mitte) und Kamerabilder von drei Bordkameras der Roboter sowie Auswahlfenster der abonnierten Topics (o.I.)

kein definierter Einsatzzweck (Use Case) der ROS-Entwickler; daher verhalten sich die Standard-Pakete bei einem solchen Einsatz nicht „ideal“. So erfolgt die Wegplanung und Kollisionsvermeidung nach den gleichen Regeln und Verfahren wie bei einem Einzel-Robotersystem. Im Multi-Roboterbetrieb kommt es allerdings in vielen Situationen zu Problemen. So werden andere, von den Sensoren nicht erfasste Roboter bei der Routenplanung nicht beachtet, obwohl deren Positionen theoretisch bekannt sind. Exemplarisch ist in Abb. 7 links die Situation zweier aufeinander zufahrender Roboter dargestellt. Jeder Roboter plant seine Route am jeweils anderen Roboter vorbei. Die Geschwindigkeit des anderen, als „statisches“ Hindernis betrachteten Roboters, wird in bei der Wegplanung somit vernachlässigt. Da nicht frühzeitig ausgewichen wird, ist eine Kollision oder Berührung im Zeitschritt  $t_3$  möglich. Beide Roboter werden ihre Geschwindigkeit im Schritt  $t_2$  reduzieren und im besten Fall im Schritt  $t_3$  zum Stehen kommen und dann eine Route um den jeweils anderen herum planen. Hierdurch verlängern sich die Fahrzeiten. In Abhängigkeit von Fahr- und Verarbeitungsgeschwindigkeit neuer Sensordaten kann es jedoch auch zur Kollision kommen. Kooperieren die Roboter und teilen sich gegenseitig ihre geplanten Routen mit, können diese bei der eigenen Planung berücksichtigt und frühzeitig ausgewichen werden (vgl. Abb. 7 rechts).

Das ROS-Paket „Multi\_robot\_collision\_avoidance“ [6] greift diese und andere Ideen auf. Es wurde jedoch nicht in die offiziellen ROS-Pakete aufgenommen, wird seit 2012 nicht mehr gepflegt und ist mit aktuellen ROS-Versionen nicht lauffähig. Einer der Gründe, warum das Paket nicht weiter gepflegt wird ist der hohe Portierungsaufwand, da mehrere ROS-Standard-Pakete verändert werden müssen. Deshalb wurde ein Wegpunktsystem und als Alternative ein Kostenkarten-Plugin für den Standard-Wegplanungsdienst aus Abschnitt 4.3 entwickelt.

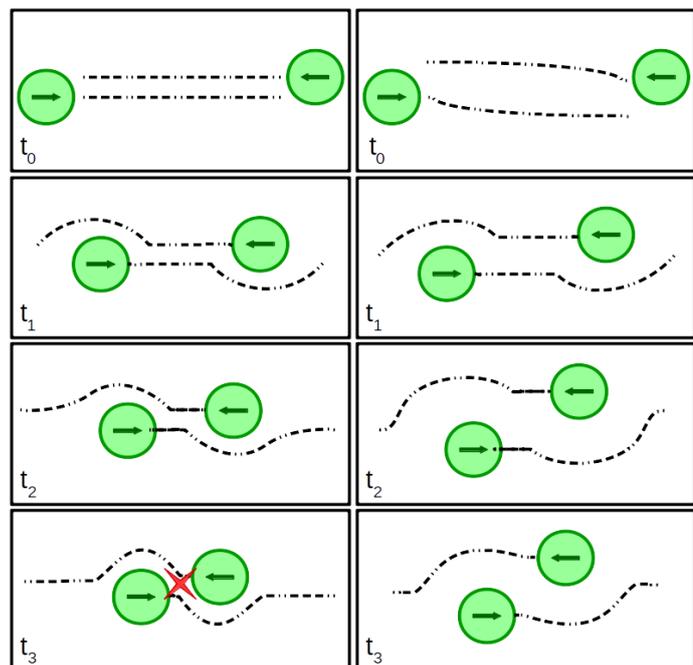


Abbildung 7: Wegplanung der Roboter bei Annahme statischer Hindernisse: links ohne Einbeziehung der Route, rechts mit Einbeziehung der Route als Gebiet erhöhter Kosten

## 5.2 Wegpunktsystem

Das Wegpunktsystem nutzt auf der zu navigierenden Fläche verteilte Wegpunkte. Ein Weg startet immer in dem Wegpunkt, der dem Roboter am nächsten ist, und führt dann von Wegpunkt

zu Wegpunkt bis zum Ziel. Ein Roboter teilt anderen Robotern seinen aktuellen und nächsten Teilpfad mit, sodass diese von ihnen nicht befahren werden. In Abb. 8 sind beispielhaft ein Raum mit Wegpunkten und eine Navigation entlang einer Wegpunktkette dargestellt. Ein Wegpunkt kann mit beliebigen anderen benachbarten Wegpunkten verbunden werden. Beschränkt man die zulässigen Verknüpfungen systematisch, so kann man beispielsweise Kreisverkehre oder Einbahnstraßen realisieren. Ein großes Problem sind

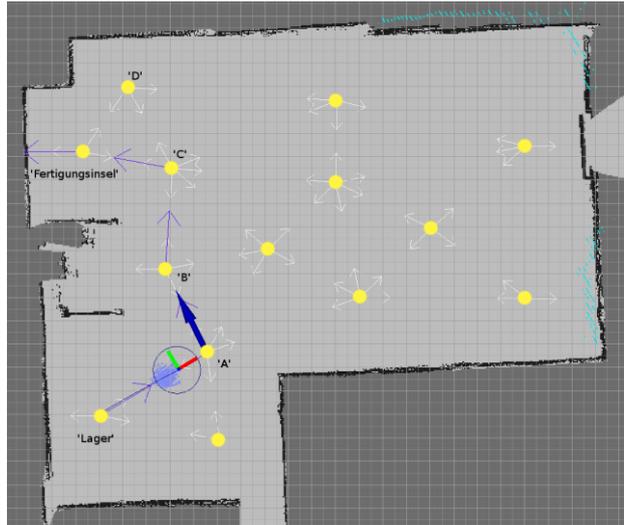


Abbildung 8: Mittels SLAM erzeugte Karte mit ergänzten Wegpunkten

durch abgestellte Güter versperrte Wegpunkte. Ist ein Wegpunkt versperrt, so kann dieser nicht mehr angefahren werden. Möglicherweise kann ein Roboter aber über andere Wegpunkte zum Ziel gelangen. Jedoch kann von Menschen ohne Kenntnis der Lage von Wegpunkten und deren Verknüpfung in der Halle nicht abgeschätzt werden, ob abgestellte Güter das autonome Transportsystem behindern. Die Kosten zwischen Wegpunkten werden oft proportional zum euklidischen Abstand und im gesperrten Zustand gleich den maximal möglichen Kosten gewählt. Zum Ermitteln des Pfades mit geringsten Kosten wird der Dijkstra-Algorithmus gewählt.

### 5.3 Erweiterte Kostenkarten

Wegen der konzeptbedingten Nachteile des Wegpunktansatzes wurde als Alternative der Standard-Wegplanungsdienst aus Abschnitt 4.3 erweitert. Dort wurde erläutert, dass die Wegplanung eine oder mehrere Teilkarten bzw. eine aus mehreren Ebenen bestehende Kostenkarte verwendet. Diese Ebenen können über ein standardisiertes Verfahren erweitert werden ohne Softwareanpassungen an den ROS-Standardpaketen vorzunehmen zu müssen. Durch die neuen Ebenen können Kosten nach Belieben erhöht oder verringert werden.

Der erste Schritt ist der Austausch der Roboterpositionen. Dadurch kann ein Roboter die Position der anderen Roboter in der Karte als Gebiet mit höheren Kosten einzeichnen und zwar unabhängig davon, ob er diese sensorisch erfassen kann. Die Größe des Roboters wird skaliert: Ist ein Roboter nah, entspricht das Gebiet erhöhter Kosten seiner wahren Größe; mit steigender Entfernung wird es kleiner eingezeichnet. Diese Vorgehensweise wurde gewählt, damit Engstellen, die nur etwas breiter sind als zwei Roboter, von diesen passiert werden können. Ein Roboter sucht in einer Engstelle die Route mit dem maximalen Abstand zu allen Hindernissen, also die Mitte. Der Freiraum, der links und rechts verbleibt, ist dann schmaler als ein Roboter

anderen Roboter unpassierbar erscheint. Wird ein entfernter Roboter jedoch zuerst kleiner in breit ist, sodass die Engstelle für einen der Karte eingezeichnet, so kann ein zweiter Roboter eine Passage planen, und der vorhandene Platz wird gleichmäßig auf beide aufgeteilt. Sind die Roboter nah beieinander sorgt die Verwendung der realen Größe für eine Kollisionsvermeidung. Zusätzlich werden die Geschwindigkeiten der Roboter mittels der Positionsänderungen durch einen Kalman-Filter geschätzt. In Richtung des Geschwindigkeitsvektors werden dann die Kosten erhöht. Diese einfache Abschätzung der zukünftigen Position eines Roboters funktioniert auch wenn eine geplante Route nicht verfügbar ist, beispielsweise wenn ein Roboter von einem Operator ferngesteuert wird. Ihr Nachteil liegt in der einfachen linearen Bewegungsprädiktion. Ein konisches „Aufweiten“ der Kostenbelegung ist dabei nicht sinnvoll, da es das Passieren und Navigieren in Engstellen nahezu unmöglich macht. Stattdessen teilt ein Roboter seine geplante Route den anderen Robotern mit und es wird auf das Ergänzen von Zusatzkosten in Folge des Geschwindigkeitsvektors verzichtet. Da Routen zyklisch neu geplant werden (in der Modellfabrik mit 3 Hz), kommt es sukzessive zu einer immer besseren Routenplanung, sollten zwei Roboter das gleiche Gebiet befahren.

Bei der Strategiekomponente des „vorausschauenden Sperrens“ wird die Konservativität der Wegplanung reduziert. Die Grundidee ist, nur diejenigen Roboterpositionen und Routenabschnitte zu sperren, die in unmittelbarer Zukunft zu erreichen sind und welche ein anderer Roboter zu der Zeit wahrscheinlich auch befahren möchte. Am besten ist das Problem an Hand von zwei hintereinanderfahrenden Robotern mit gleicher Geschwindigkeit zu zeigen: Obwohl der hintere den vorausfahrenden Roboter nicht einholen kann plant er eine Route, um diesen zu überholen und blockiert damit die Durchfahrt für andere Roboter, welche die Passage in entgegengesetzter Richtung nutzen oder beide Roboter überholen wollen.

#### **5.4 Auftragsverteilung und Kooperation**

Beim Einsatz mehrerer Roboter müssen Aufträge sinnvoll verteilt werden. Einfache Heuristiken wie „freien Roboter mit minimalem Abstand zum Ziel“ können dafür ausreichen. Bei sehr kurzen Strecken kann die Streckenlänge vernachlässigt werden, wenn die meiste Zeit für das An- und Abdocken von den Stationen benötigt wird. In der Modellfabrik gibt es bei der Vergabe von Aufträgen noch einen zweiten Aspekt: Transportaufträge benötigen einen großen Becher. Diese Aufträge können nur an Roboter mit großen Bechern vergeben werden. Ein weiterer Aspekt ist die Kooperation beim Belegen von Roboter-Docks. Ist ein Dock belegt, so fährt ein weiterer Roboter, der ebenfalls einfahren möchte, zuerst auf eine Warteposition, bis der erste Roboter

das Dock freigibt, siehe beispielsweise Abb. 10 o.r. So behindert er nicht die Ausfahrt des Ersten. Dabei gilt in der Modellfabrik „first come, first served“: Der Roboter, der sich als erstes einem Dock nähert, fährt zur Einfahrstation und der zweite Roboter zur Warteposition.

## 6 Transportrobotereinsatz in Modellfabrik

In der Modellfabrik  $\mu$ Plant (vgl. Abb. 9) transportieren die mobilen Roboter große und kleine Becher (vgl. Abb. 2), um den Materialfluss zwischen den einzelnen Produktionseinrichtungen zu realisieren. Das Material wird in den bereits angesprochenen Roboter-Docks aufgenommen oder abgegeben. Mechanische Leitplanken und drei Infrarot-Leitstrahlen sorgen für ein ausreichend genaues Erreichen der Endposition im Dock (vgl. Abb. 10 u.l.), sodass die Hebevorrichtungen der Abfüll- und Entleerstationen (vgl. Abb. 10 u.r.) bzw. das Greifwerkzeug des Industrieroboters im Hochregallager (vgl. Abb. 10 o.l.) ohne Zusatzmaßnahmen die Becher sicher greifen bzw. die Abfüllvorrichtungen die Becher sicher befüllen können.

Mit dem Aufbau des Ein-Robotersystems wurde 2014 begonnen und es wurde im Juli 2015 auf ein Multi-Robotersystem erweitert. Seit September 2016 läuft der Betrieb der Gesamtfabrik. Dabei sind immer drei Transportroboter operativ und einer dient als Reserve. Der Einsatz der mobilen Roboter in der Modellfabrik  $\mu$ Plant ist exemplarisch in einem Video festgehalten [9].

Beim Ein-Robotersystem hat sich gezeigt, dass eine hohe Positioniergenauigkeit merkmalsreiche Räume mit klaren Strukturen erfordert. Daher wurde die Prozessinsel I mit einem Sichtschutz auf Sensorhöhe versehen, um die unregelmäßige und bewegliche Verschlauchung zu verdecken. Auch die Bedeutung des Grips der Räder wurde in Bezug auf Verschmutzung und schwere Beladung in Verbindung mit glatten Böden zuerst unterschätzt. Deshalb wurden die originalen Castorräder verkleinert, um den Anpressdruck der Antriebsräder zu erhöhen. Weiterhin wurde das häufige Schreiben der RFID-Tags an den Transportbehältern in einer Umgebung mit viel Metall und Wasser unterschätzt. Daher wurden alle RFID-Systeme mit einer zweiten Antenne versehen. Der Aufbau des Multi-Robotersystems erforderte ein



Abbildung 9: Modellfabrik  $\mu$ Plant: Im Vordergrund ist der Leitstand, im Hintergrund sind die verschiedenen Produktionseinrichtungen (Prozessinseln, Fertigungszelle, Abfüll- und Entleerstationen, Hochregallager) zu sehen

Mehrkanal-WLAN, da eine intensive WLAN-Kommunikation bei nur einem Kanal die Datenpaketlaufzeit stark erhöht. Mit den beschriebenen Verbesserungen ließ sich ein störungsfreier Betrieb der Modellfabrik erreichen.

## 7 Zusammenfassung und Ausblick

Autonome Mobile Roboter (AMR) erfahren großes Interesse, da sie flexibler als fahrerlose Transportsysteme an geänderte Produktionsbedingungen angepasst werden und ungeplante Hindernisse umfahren können. Im Beitrag wurde die Navigation von AMR erläutert und Erweiterungen um kooperative Strategiekomponenten für den Betrieb von Multi-Robotersystemen vorgestellt. Das Multi-Robotersystem wird für Transportaufgaben in der Modellfabrik  $\mu$ Plant eingesetzt, in denen es sich im Testbetrieb nach der Umsetzung verschiedener Verbesserungen bewährt hat. Für die Zukunft sind weitere Detailverbesserungen vorgesehen. Ein Einsatz der Modellfabrik als Industrie 4.0-Testumgebung und für andere Aufgaben in Forschung, Lehre und gewerblicher Weiterbildung sind geplant.



Abbildung 10: o.l.: Produkteinlagerung; o.r.: Roboter fährt in Station ein, um Produkt abgefüllt zu bekommen; u.l.: Warteschlangenbildung vor Station; u.r.: Entleeren von großem Becher durch Hebevorrichtung

## 8 Literaturverzeichnis

- [1] Claes, D.: Collision Avoidance under Bounded Localization Uncertainty. M. Sc. Thesis, Department of Knowledge Engineering, Maastricht University, Maastricht, the Netherlands, August 2012.
- [2] Enright, J.; Wurman, P. R.: Optimization and Coordinated Autonomy in Mobile Fulfillment Systems. In Proceedings of the 9th AAAI Conference on Automated Action Planning for Autonomous Mobile Robots (AAAIWS'11-09). AAAI Press. 2011, S. 33-38.
- [3] Habib, A.; Bonow, G.; Kroll, A.; Hegenberg, J.; Schmidt, L.; Barz, T.; Schulz, D.: Forschungsprojekt RoboGasInspector: Gaslecksuche mit autonomen mobilen Robotern; Technische Sicherheit, Vol. 3(5), S. 10-15, 2013
- [4] Hegenberg, J.; Herrmann, R.; Ziegner, D.; Schmidt, L.; Guenther, T.; Müller, A.O.; Kroll, A.; Barz, T.; Schulz, D.: Forschungsprojekt Robot}Air{: Praxistaugliches Boden-Luft-Servicerobotersystem für die Inspektion industrieller Druckluftversorgung und die Verbesserung der Arbeitsumgebungs-faktoren; Technische Sicherheit, Vol. 5 (5) , S. 16-22, 2015
- [5] Kobuki-Manual [online] [Zugriff am 10.04.18]. Verfügbar unter:  
<http://kobuki.yujinrobot.com/wiki/online-user-guide/>
- [6] Kroll, A., Dürrbaum, A.; Arengas, D.; Jäschke, B.; Al Mawla, H.; Geiger, A.:  $\mu$ Plant: Model factory for the automatization of networked, heterogeneous and flexibly changeable multi-product plants; Automation 2016, VDI-Berichte 2284, Baden-Baden, 2016
- [7] Kroll, A.; Dürrbaum, A.; Arengas, D.; Al Mawla, H.; Kistner, L.; Rehmer, A.:  $\mu$ Plant: Eine automatisierungstechnisch-orientierte Modellfabrik für vernetzte heterogene Systeme; atp edition, Vol. 59 (9), S. 40-53, September 2017
- [8] Liang, C., et al.: Automated robot picking system for e-commerce fulfillment warehouse application, 14th IFToMM World Congress, 2015.
- [9] Modellfabrik  $\mu$ Plant [online, Video] [Zugriff am 10.04.18]. Verfügbar unter:  
<https://www.youtube.com/watch?v=gObxcGxFDG4>
- [10] Nehmzov, U.: Mobile Robotik – Eine praktische Einführung, Springer, 2002.
- [11] Overmeyer, L., et al.: Intelligente Flurförderzeuge durch die Implementierung kognitiver Systeme. In: B. Vogel-Heuser et al. (Hrsg.), Handbuch Industrie 4.0, Bd. 3, 2. Aufl., Springer, 2017.
- [12] Referenzplattform Turtlebot2 [online] [Zugriff am 10.04.18]. Verfügbar unter:  
<http://www.turtlebot.com/turtlebot2/>
- [13] Richtlinie VDI 2510:2005-10, Fahrerlose Transportsysteme (FTS), Bl. 1-3, 2005-2013.
- [14] Richtlinie ISO 8373:2012-3, Robots and robotic devices, 2012.
- [15] Ulrich, G.: Fahrerlose Transportsysteme, Springer-Vieweg, 2014.